

Nama : Saphira Nuria Salsabila
NIM : 22305141050
Kelas : Matematika B

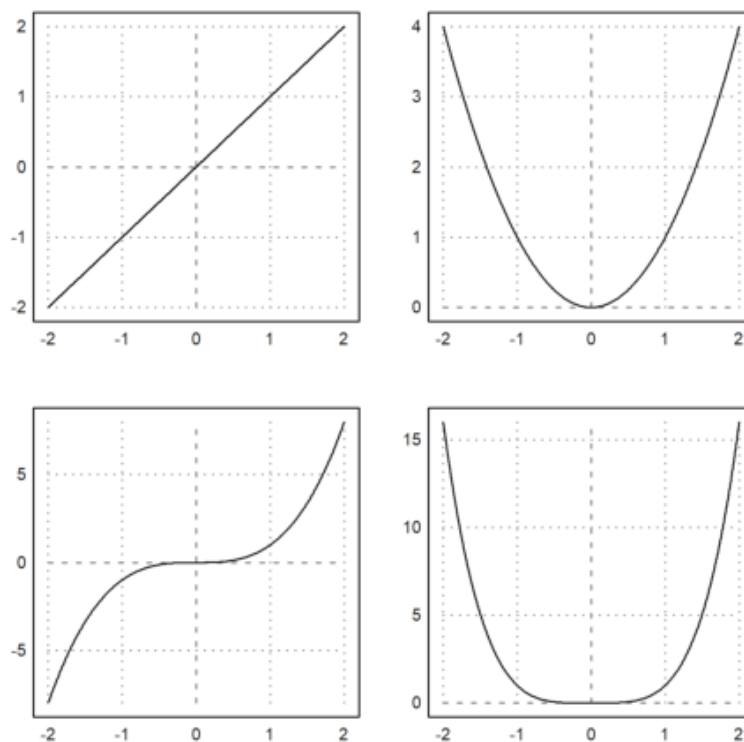
Subtopik 5 - Menggambar Beberapa Kurva Sekaligus

Dalam subtopik ini, kita akan membahas cara menggambar menggambar kurva sekaligus (dalam satu baris perintah). Dengan adanya materi pada subtopik ini, diharapkan kita bisa menggambar beberapa kurva dalam bidang koordinat yang berbeda dalam satu waktu. Untuk melakukannya, kita dapat menggunakan perintah `figure()`. Berikut contoh menggambar beberapa kurva sekaligus.

Gambarkan plot fungsi berikut:

$$x^n, 1 \leq n \leq 4$$

```
>reset;  
>figure(2,2); ...  
>for n=1 to 4; figure(n); plot2d("x^"+n); end;...  
>figure(0):
```



Penjelasan dari plot fungsi

$$x^n, 1 \leq n \leq 4$$

- reset;

Perintah ini digunakan sebagai awalan/permulaan menggambar grafik baru. Perintah ini juga berfungsi untuk menghapus grafik yang telah ada sebelumnya sehingga dapat memulai kembali untuk membuat grafik yang baru.

- figure(2x2);

Perintah figure() digunakan untuk membuat jendela grafik dengan ukuran axb di mana a merupakan banyaknya baris dan b merupakan banyaknya kolom. Dalam hal ini perintah figure(2,2) berarti akan dibuat jendela grafik berukuran 2x2 (2 baris dan 2 kolom).

- for n=1 to 4;

Perintah ini berfungsi untuk melakukan looping, dalam hal ini sebanyak 4 kali yaitu dari 1 sampai 4.

- figure(n);

Perintah ini berfungsi untuk beralih dari grafik satu ke grafik lainnya (grafik ke-n)

- plot2d("x^"+n);

Perintah plot2d() berfungsi untuk membuat plot fungsi matematika. Dalam hal ini fungsi yang diplot adalah x^n , di mana n adalah nilai dari variabel yang dikenai looping. Sehingga, akan ditampilkan plot dari x^1 , x^2 , x^3 , dan x^4 pada bidang grafik.

- end;

Perintah ini menunjukkan berakhirnya looping;

- figure(0);

Perintah ini berfungsi untuk beralih kembali ke jendela grafik utama.

```
>figure(2,2); ...  
>for n=1 to 4; figure(n); plot2d(x^n); end;
```

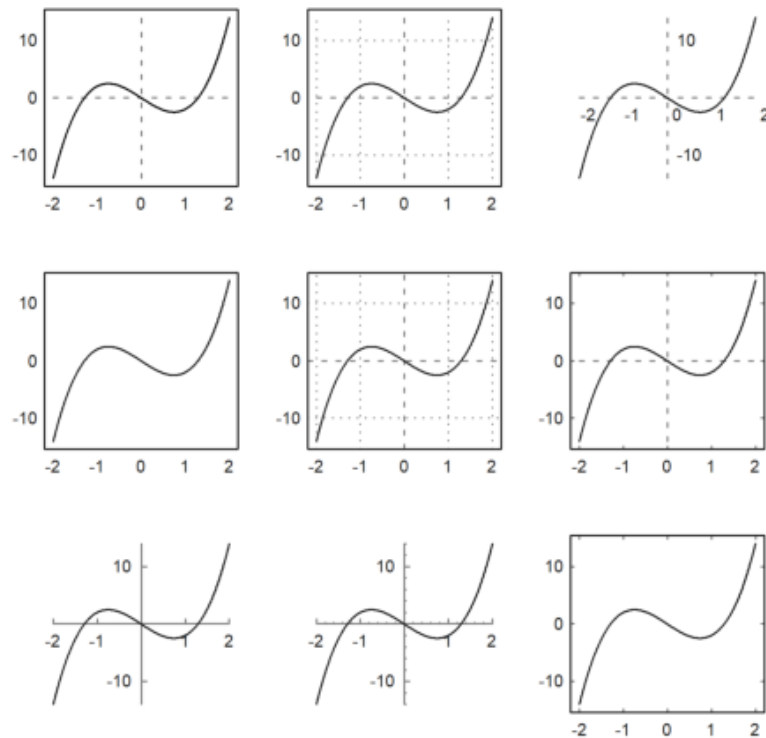
```
Variable x not found!  
Error in ^  
Error in:  
figure(2,2); for n=1 to 4; figure(n); plot2d(x^n); end; ...  
^
```

Dari masalah di atas, kita tahu bahwa untuk membuat kurva fungsi x^n tidak dapat semerta-merta langsung diketik x^n pada baris perintah, melainkan harus diketik dengan ("x^"+n). Tanda petik dua ("...") berfungsi untuk mengidentifikasi ekspresi matematika. Sedangkan tanda plus (+) berfungsi untuk menggabungkan string dengan variabel yang ada.

Contoh Soal :
Gambarkan plot fungsi

$$f(x) = 3x^3 - 5x; -2 \leq x \leq 2$$

```
>reset;  
>figure(3,3); ...  
>for k=1:9 figure(k); plot2d("3x^3-5x",-2,2,grid=k); end; ...  
>figure(0) :
```



Penjelasan dari plot fungsi

$$f(x) = 3x^3 - 5x, -2 \leq x \leq 2$$

- reset;

Perintah ini berfungsi untuk memulai ulang perintah, dengan menghapus grafik yang telah ada sebelumnya dan membuat perintah baru.

- figure(3,3);

Perintah ini berfungsi untuk membuat jendela grafik dengan ukuran 3x3. Sehingga nantinya akan ada empat macam jendela grafik yang akan ditampilkan dengan tata letak 3 baris dan 3 kolom.

- for k=1:9;

Perintah ini berfungsi untuk melakukan looping perintah sebanyak 9 kali.

- figure(k);

Perintah ini berfungsi untuk beralih dari jendela grafik satu ke jendela grafik lainnya (grafik ke-k).

- plot2d("3x^3-5x,-2,2,grid=k");

Perintah plot2d() berfungsi untuk membuat plot fungsi matematika. Dalam hal ini fungsi yang diplot adalah $3x^3 - 5x$, dengan batas sumbu x dari -2 sampai 2. Argumen grid=k berfungsi untuk mengaktifkan grid pada jendela grafik ke-k.

- end;

Perintah ini menunjukkan berakhirnya proses looping

- figure(0);

Perintah ini berfungsi untuk kembali ke jendela grafik utama.

Dari contoh di atas dapat kita perhatikan bahwa tampilan plot dari yang ke-1 hingga ke-9 memiliki tampilan yang berbeda-beda. EMT memiliki berbagai gaya plot 2D yang dapat dijalankan menggunakan perintah grid=n di mana n adalah jumlah langkah minimal. Setiap nilai n memiliki tampilan plot adaptif yang berbeda dalam plot 2D, di antaranya yaitu:

0 : tidak ada grid (kisi), frame, sumbu, dan label, hanya kurva saja

1 : dengan sumbu, label-label sumbu di luar frame jendela grafik

2 : tampilan default

3 : dengan grid pada sumbu x dan y, label-label sumbu berada di dalam jendela grafik

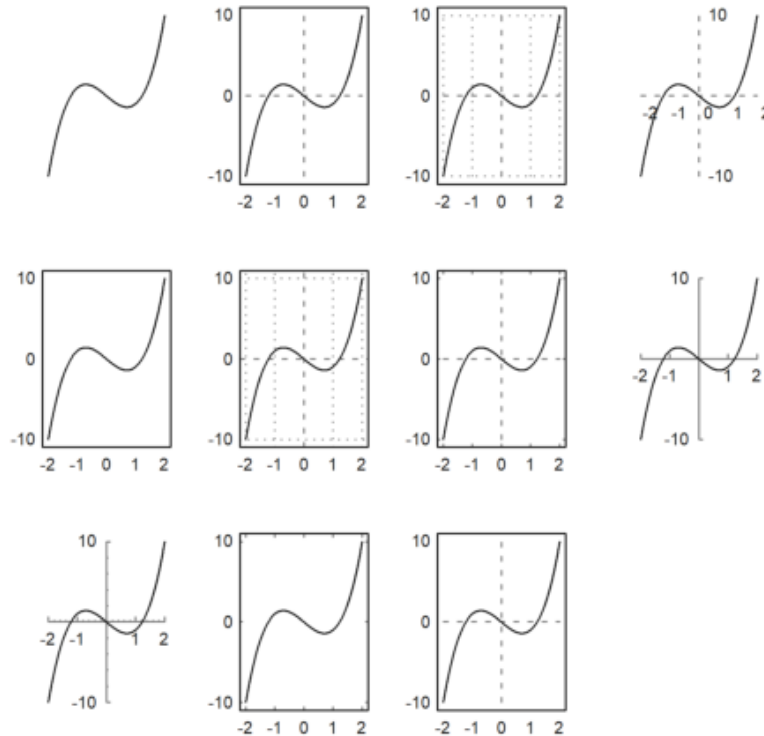
- 4 : tidak ada grid (kisi), sumbu x dan y, dan label berada di luar frame jendela grafik
 - 5 : tampilan default tanpa margin di sekitar plot
 - 6 : hanya dengan sumbu x y dan label, tanpa grid
 - 7 : hanya dengan sumbu x y dan tanda-tanda pada sumbu.
 - 8 : hanya dengan sumbu dan tanda-tanda pada sumbu, dengan tanda-tanda yang lebih halus pada sumbu.
 - 9 : tampilan default dengan tanda-tanda kecil di dalam jendela
 - 10: hanya dengan sumbu-sumbu, tanpa tanda
-

Contoh Soal:

Gambarkan plot fungsi berikut :

$$g(x) = 2x^3 - 3x$$

```
>reset;  
>aspect();  
>figure(3,4);  
>figure(1); plot2d("2x^3-3x", grid=0);...  
>figure(2); plot2d("2x^3-3x", grid=1);...  
>figure(3); plot2d("2x^3-3x", grid=2);...  
>figure(4); plot2d("2x^3-3x", grid=3);...  
>figure(5); plot2d("2x^3-3x", grid=4);...  
>figure(6); plot2d("2x^3-3x", grid=5);...  
>figure(7); plot2d("2x^3-3x", grid=6);...  
>figure(8); plot2d("2x^3-3x", grid=7);...  
>figure(9); plot2d("2x^3-3x", grid=8);...  
>figure(10); plot2d("2x^3-3x", grid=9);...  
>figure(11); plot2d("2x^3-3x", grid=10);...  
>figure(0):
```



Penjelasan dari plot fungsi:

$$g(x) : 2x^3 - 3x$$

- `aspect(1, 2);`

Perintah `aspect()` berfungsi untuk mengatur rasio aspek dari jendela grafik. Dalam hal ini, perintah `aspect(1, 2);` akan menghasilkan plot dengan perbandingan panjang sumbu x dan sumbu y berturut-turut 1:2.

- `figure (3,4);`

Perintah ini digunakan untuk membuat jendela grafik dengan ukuran 3x4 (3 baris dan 4 kolom).

- `figure(1); plot2d("2x^3-3x", grid(0));...`

Merupakan perintah untuk beralih ke jendela grafik ke-1 dan menggambar plot dari fungsi $2x^3-3x$ tanpa grid, frame, maupun sumbu.

- `figure(2); plot2d("2x^3-3x",grid=1); ...`

Merupakan perintah untuk beralih ke jendela grafik kedua dan menggambar plot dari fungsi $2x^3-3x$ dengan grid hanya pada sumbu x dan y.

- `figure(3); plot2d("2x^3-3x",grid=2); ...`

Merupakan perintah untuk beralih ke jendela grafik ketiga dan menggambar plot dari fungsi $2x^3-3x$ dengan tampilan default, termasuk tanda-tanda default pada sumbu.

- `figure(4); plot2d("2x^3-3x",grid=3); ...`

Merupakan perintah untuk beralih ke jendela grafik keempat dan menggambar plot dari fungsi $2x^3-3x$ dengan grid pada sumbu x dan y, serta label-label sumbu yang ada di dalam jendela.

- `figure(5); plot2d("2x^3-3x",grid=4); ...`

Merupakan perintah untuk beralih ke jendela grafik kelima dan menggambar plot dari fungsi $2x^3-3x$ tanpa tanda-tanda sumbu, hanya label-label yang ada.

- `figure(6); plot2d("2x^3-3x",grid=5); ...`

Merupakan perintah untuk beralih ke jendela grafik keenam dan menggambar plot dari fungsi $2x^3-3x$ dengan tampilan default, tetapi tanpa margin di sekitar plot.

- `figure(7); plot2d("2x^3-3x",grid=6); ...`

Merupakan perintah untuk beralih ke jendela grafik ketujuh dan menggambar plot dari fungsi $2x^3-3x$ hanya dengan sumbu-sumbu (tanpa grid atau label).

- figure(8); plot2d("2x^3-3x",grid=7); ...

Merupakan perintah untuk beralih ke jendela grafik kedelapan dan menggambar plot dari fungsi $2x^3-3x$ hanya dengan sumbu-sumbu dan tanda-tanda pada sumbu.

- figure(9); plot2d("2x^3-3x",grid=8); ...

Merupakan perintah untuk beralih ke jendela grafik kesembilan dan menggambar plot dari fungsi $2x^3-3x$ hanya dengan sumbu-sumbu dan tanda-tanda pada sumbu, dengan tanda-tanda yang lebih halus pada sumbu.

- figure(10); plot2d("2x^3-3x",grid=9); ...

Merupakan perintah untuk beralih ke jendela grafik kesepuluh dan menggambar plot dari fungsi $2x^3-3x$ dengan tanda-tanda default kecil di dalam jendela.

- figure(11); plot2d("2x^3-3x",grid=10); ...

Merupakan perintah untuk beralih ke jendela grafik kesebelas dan menggambar plot dari fungsi $2x^3-3x$ hanya dengan sumbu-sumbu, tanpa tanda-tanda.

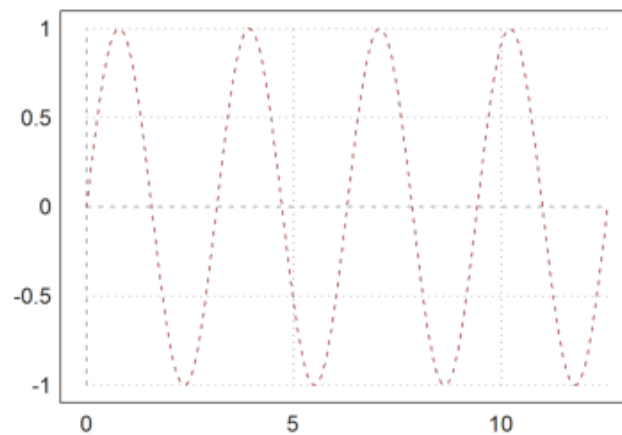
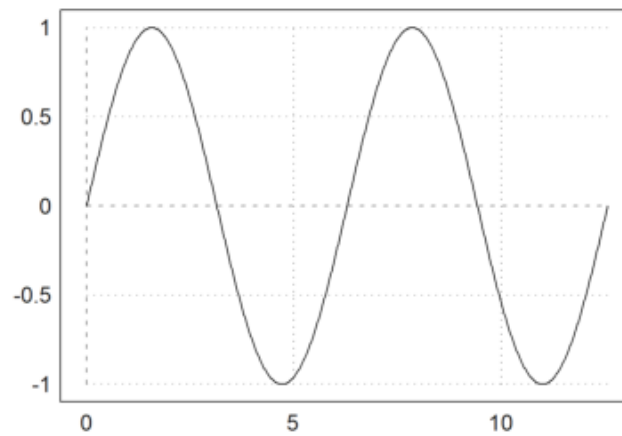
- figure(0);

Merupakan perintah untuk beralih kembali ke jendela grafik utama atau jendela grafik dengan nomor 0 setelah semua perintah dalam urutan selesai dieksekusi.

Dari ketiga contoh di atas, dapat disimpulkan bahwa untuk menggambar beberapa kurva sekaligus, dapat dilakukan dengan satu baris perintah ataupun dengan cara mendefinisikan 1 per 1.

Terlihat beberapa jenis grid memiliki tampilan yang mirip atau sama, seperti 1 dan 2, 2 dan 5, 4 dan 9, serta 7, dan 8. Untuk dapat membedakannya secara lebih jelas, kita akan mengubah grid dari contoh berikut.

```
>reset;
>aspect(2,3);
>figure(2,1);...
>figure(1); plot2d("sin(x)",0,4pi, grid=2);...
>figure(2); plot2d("sin(2x)",0,4pi, grid=2,color=red,style="--",);
>figure(0):
```



Dalam visualisasi plot, dapat dilakukan berbagai modifikasi mulai dari jenis grid, warna, style, dan lain-lain. Modifikasi tersebut akan dibahas lebih lanjut pada subtopik selanjutnya.

Contoh Soal:

Gambarkan plot fungsi-fungsi berikut:

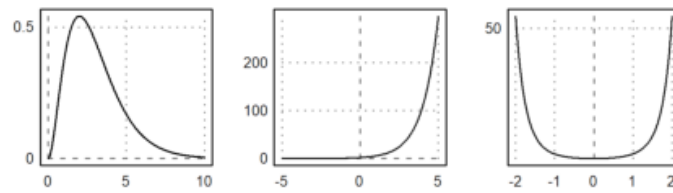
$$f(x) = x^2 e^{-x}, 0 \leq x \leq 10$$

$$g(x) = 2e^x, -5 \leq x \leq 5$$

$$h(x) = e^{x^2}, -2 \leq x \leq 2$$

```
>reset;
>aspect(3,1);
>figure(1,3);...
>figure(1); plot2d("x^2*exp(-x)",0,10);...
>figure(2); plot2d("2*exp(x)",-5,5);...
>figure(3); plot2d("exp(x^2)",-2,2);...
```

```
>figure(0) :
```

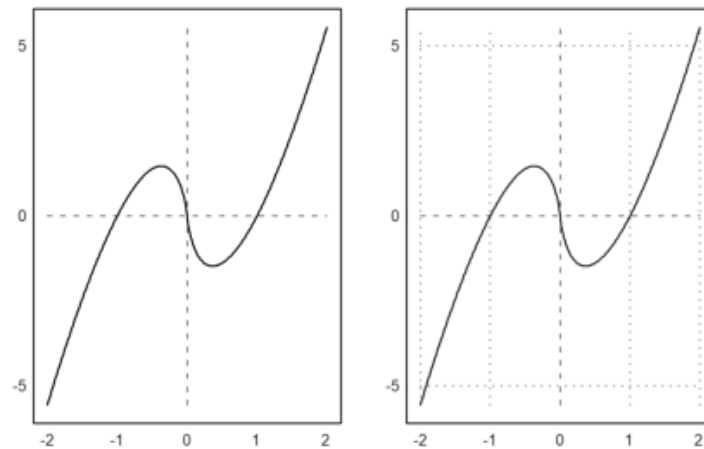


Contoh Soal:

Gambarkan plot dari fungsi berikut:

$$2x\log(x^2)$$

```
>reset;  
>aspect(3/2);  
>figure(1,2);...  
>for a=1:2 ;figure(a); plot2d("2*x*log(x^2)",-2,2,grid=a); end;...  
>figure(0) :
```



Subtopik 14 - Menggambar Daerah yang Dibatasi Beberapa Kurva

Pada subtopik sebelumnya telah kita ketahui dan pelajari bersama bahwa EMT dapat melakukan visualisasi plot mulai dari bentuk ekspresi langsung hingga plot dari fungsi-fungsi.

Subtopik ini merupakan kelanjutan dari subtopik sebelumnya, yaitu membentuk/menggambar daerah dari perpotongan beberapa kurva yang telah didefinisikan. Hal ini dapat bermanfaat untuk membantu dalam menyelesaikan permasalahan dalam matematika, salah satu contohnya seperti optimasi program linear, di mana disajikan beberapa fungsi-fungsi kendala beserta dengan fungsi tujuannya dan perlu divisualisasikan dalam bentuk grafik untuk melihat dimana letak daerah layak untuk menentukan nilai optimum.

Dalam EMT ada beberapa perintah yang digunakan untuk menggambar daerah yang dibatasi oleh beberapa kurva, di antaranya yaitu:

- plot2d

Digunakan untuk melakukan plotting.

- filled=true

Digunakan untuk memberikan isian/arsiran pada daerah/area di bawah kurva saat plotting.

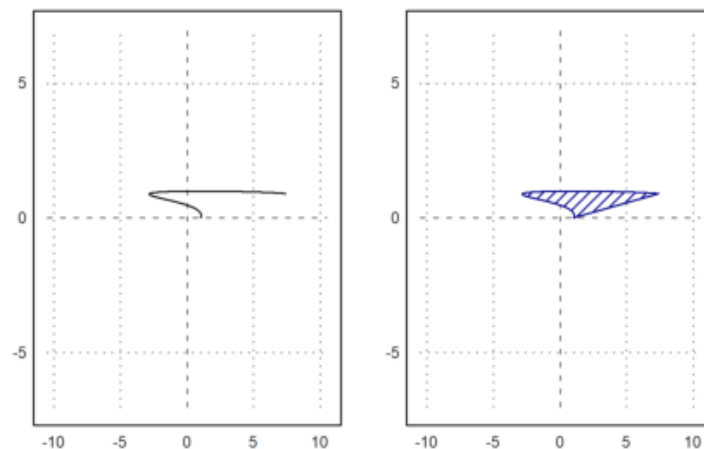
- style="..."

Digunakan untuk memilih gaya kurva yang akan digunakan saat plotting. Anda dapat memilih dari beberapa gaya, seperti ":", "/", "\", atau "-". Dan hal ini mempengaruhi tampilan daerah kurva yang terbentuk.

- fillcolor

Digunakan untuk menentukan warna isian yang akan digunakan untuk mengisi area di bawah kurva.

```
>t=linspace(0, 2pi, 1000); //parameter untuk kurva
>x=cos(t)*exp(t/pi); y=sin(t/pi); //x(t) dan y(t)
>figure(1,2); aspect(3/2)
>figure(1); plot2d(x,y,r=7); //plot kurva
>figure(2); plot2d(x,y,r=7,>filled,style="/",fillcolor=blue); //mengisi kurva
>figure(0):
```



Penjelasan

- t=linspace(0,2pi,1000);

Pada langkah pertama yaitu mendefinisikan parameter t sebagai serangkaian 1000 titik antara 0 dan 2pi. Parameter t ini akan digunakan sebagai parameter untuk menggambar kurva.

- cos(t)*exp(t/pi); y=sin(t)*exp(t/pi);

Didefinisikan dua vektor x dan y yang merupakan koordinat x dan y dari kurva yang akan digambar.

Fungsi :

$$\cos(t) * \exp(t/\pi)$$

digunakan untuk menghitung komponen $x(x(t))$, dan

$$\sin(t) * \exp(t/\pi)$$

digunakan untuk menghitung komponen $y(y(t))$ dari kurva.

- `figure(1,2); aspect(3/2)`

Perintah ini digunakan untuk mengatur tampilan gambar. Perintah `figure(1,2)` digunakan membuat dua gambar (1 dan 2) dalam satu jendela gambar. Dan perintah `aspect(3/2)` mengatur rasio aspek gambar menjadi 3:2, yang mempengaruhi bentuk dan ukuran gambar yang akan digambar.

- `figure(1); plot2d(x,y,r=7);`

Perintah ini memilih gambar pertama (1) dan menggunakan perintah `plot2d` untuk menggambar kurva yang dihitung sebelumnya. Parameter `r=10` mengatur lebar garis plot. Ini menghasilkan kurva tanpa adanya isi atau arsiran di dalamnya.

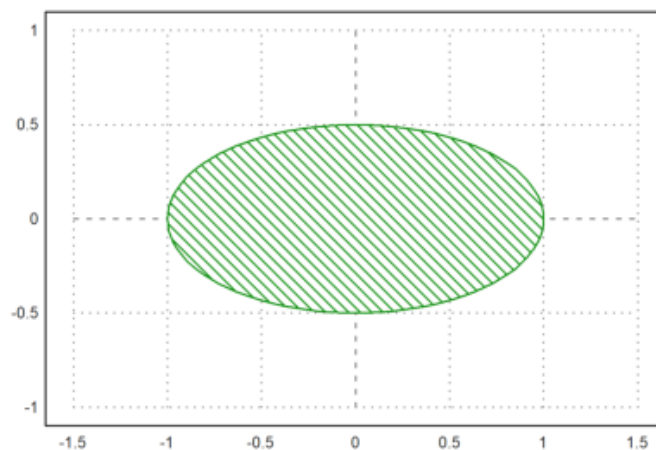
- `figure(2); plot2d(x,y,r=7,>filled,style="\/",fillcolor=blue);`

Selanjutnya pada perintah ini beralih ke gambar kedua (2) dan menggunakan perintah `plot2d` lagi untuk menggambar kurva yang sama dengan pengisian area di bawahnya. Perintah `>filled` digunakan untuk mengisi area di bawah kurva, `style="\/"` digunakan untuk mengatur gaya garis menjadi garis miring, dan `fillcolor=blue` digunakan untuk mengatur warna isian menjadi biru.

- `figure(0);`

Baris perintah ini digunakan untuk mengakhiri gambar dan kembali ke tampilan biasa tanpa gambar. Perintah ini berfungsi untuk menyelesaikan proses penggambaran.

```
>x=linspace(0,2pi,100); plot2d(cos(x),sin(x)*0.5,r=1,>filled,style="\"):
```



Penjelasan:

- `x=linspace(0,2pi,100);`

Mendefinisikan vektor x dengan menggunakan perintah `linspace`. `linspace` digunakan untuk membuat vektor dengan 100 titik yang secara merata tersebar antara 0 dan 2π . Dalam konteks ini, vektor x akan digunakan sebagai parameter saat menggambar kurva.

- `plot2d(cos(x),sin(x)*0.5,r=1,>filled,style="\/"):`

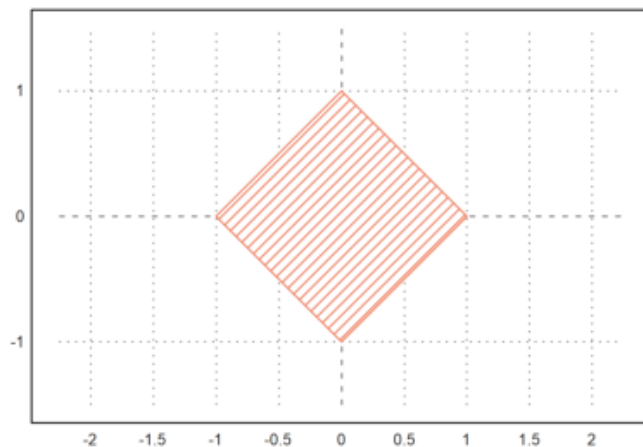
Ini merupakan perintah utama yang digunakan untuk menggambar plot. Perintah ini memiliki beberapa parameter sebagai berikut:

> `cos(x)` adalah komponen x dari kurva. Ini adalah hasil dari fungsi kosinus yang dihitung pada vektor x .

> `sin(x)*0.5` adalah komponen y dari kurva. Ini adalah hasil dari fungsi sinus yang dihitung pada vektor x dan kemudian dikalikan dengan 0,5, yang mengubah amplitudonya.

- > `r=1` mengatur lebar garis plot menjadi 1.
- > `filled` digunakan untuk mengisi area di bawah kurva, sehingga menciptakan daerah yang terisi.
- > `style="/"` mengatur gaya garis kurva menjadi garis miring kekiri dengan menggunakan tanda backslash(`\`)

```
>t=linspace(0,2pi,4);
>plot2d(cos(t),sin(t),>filled, style="/",fillcolor=orange,r=1.5):
```



Penjelasan:

- `t=linspace(0,2pi,4);`

Pada perintah ini, kita definisikan vektor `t` dengan menggunakan perintah `linspace`. `Linspace` digunakan untuk membuat vektor dengan 4 titik yang terletak secara merata antara 0 dan 2π . Dalam konteks ini, vektor `t` akan digunakan sebagai parameter saat menggambar kurva.

- `plot2d(cos(t),sin(t),>filled,style="/",fillcolor=orange,r=1.2):`

Perintah ini merupakan perintah utama yang digunakan untuk menggambar plot.

Perintah ini memiliki beberapa parameter sebagai berikut:

- > `cos(t)` adalah komponen x dari kurva.

- > `sin(t)` adalah komponen y dari kurva.

- > `filled` digunakan untuk mengisi area di bawah kurva, sehingga menciptakan bentuk yang terisi. Ini berarti daerah di bawah kurva akan diwarnai.

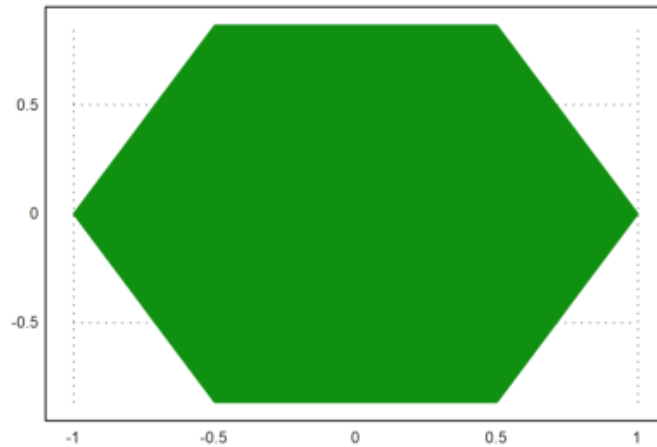
- > `style="/"` mengatur gaya garis kurva menjadi garis miring ("`/`").

- > `fillcolor=orange` mengatur warna isian daerah di bawah kurva menjadi oranye.

- > `r=1.2` mengatur lebar garis plot menjadi 1.2.

```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):

```



Penjelasan:

- `t=linspace(0,2pi,6);`

Pada perintah ini, kita definisikan vektor `t` dengan menggunakan perintah `linspace`. `Linspace` digunakan untuk membuat vektor dengan 6 titik yang terletak secara merata antara 0 dan 2π . Dalam konteks ini, vektor `t` akan digunakan sebagai parameter saat menggambar kurva.

- `plot2d(cos(t),sin(t),>filled,style='')`:

Ini adalah perintah utama yang digunakan untuk menggambar plot. Perintah ini memiliki beberapa parameter sebagai berikut:

> `cos(t)` adalah komponen x dari kurva.

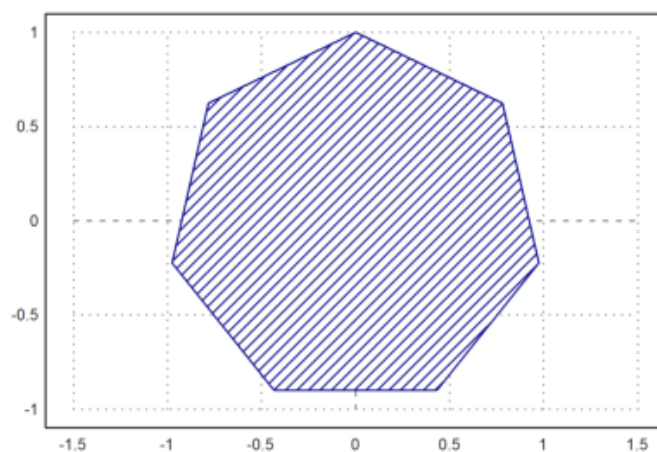
> `sin(t)` adalah komponen y dari kurva.

> `filled` digunakan untuk mengisi area di bawah kurva, sehingga menciptakan bentuk yang terisi. Ini berarti daerah di bawah kurva akan diisi dengan warna atau pola tertentu.

> `style=''` mengatur isian kurva menjadi warna solid dengan menggunakan simbol tanda pagar (``)

```
>t=linspace(0,2pi,7); ...
```

```
>plot2d(sin(t),cos(t),r=1,>filled,style="/",fillcolor=blue):
```



Penjelasan:

- `t=linspace(0,2pi,7);`

Fungsi `linspace` digunakan untuk membuat array berisi sejumlah nilai yang merata dalam rentang tertentu. Dalam hal ini, rentangnya adalah dari 0 hingga 2π (dua kali nilai π) dan sebanyak 7 titik akan dihasilkan. Ini akan digunakan sebagai sudut dalam koordinat polar untuk menggambar data.

- `plot2d(sin(t),cos(t),r=1,>filled,style="//",fillcolor=blue):`

Ini adalah perintah untuk melakukan plotting data. Terdapat beberapa argumen di sini:

> `sin(t)`: Ini adalah nilai sinus dari setiap elemen dalam array `t`. Ini akan digunakan sebagai komponen sumbu X dalam koordinat polar.

> `cos(t)`: Ini adalah nilai kosinus dari setiap elemen dalam array `t`. Ini akan digunakan sebagai komponen sumbu Y dalam koordinat polar.

> `r=1`: Ini adalah argumen opsional yang menentukan radius plot. Dalam hal ini, radiusnya diatur menjadi 1.

> `filled`: Ini adalah argumen yang menginstruksikan untuk mengisi area di dalam kurva plot.

> `style="//"`: Ini adalah argumen yang menentukan gaya garis yang digunakan untuk plot. Di sini, garisnya akan berbentuk garis miring ("`//`").

> `fillcolor=blue`: Ini adalah argumen yang menentukan warna pengisian untuk area di dalam kurva plot. Dalam hal ini, warnanya diatur menjadi biru.

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);  
>plot2d(x,y,<grid,<frame,>filled):
```



Penjelasan:

- `t = linspace(0, 2*pi, 1000);`

Ini adalah perintah untuk membuat vektor `t` yang berisi 1000 nilai yang merata terdistribusi antara 0 hingga 2π . Vektor `t` ini akan digunakan sebagai parameter waktu atau sudut dalam parameterisasi lingkaran.

`linspace(0, 2*pi, 1000)` membuat 1000 titik antara 0 hingga 2π , memberikan sudut-sudut yang merata di sepanjang satu putaran lingkaran.

- `x = cos(3*t); y = sin(4*t);`

Ini adalah perintah untuk menghitung vektor `x` dan `y` yang menggambarkan lintasan dalam koordinat polar.

> `x = cos(3*t)`; menghitung nilai `x` sebagai hasil dari fungsi kosinus dari 3 kali nilai `t`. Ini akan menghasilkan osilasi yang lebih cepat pada sumbu `x`.

> `y = sin(4*t)`; menghitung nilai `y` sebagai hasil dari fungsi sinus dari 4 kali nilai `t`. Ini akan menghasilkan osilasi yang lebih cepat pada sumbu `y`.

- `plot2d(x, y, <grid, <frame, >filled);`

Ini adalah perintah untuk membuat plot dari vektor `x` dan `y`. Berikut adalah rincian perintah ini:

`x` adalah vektor yang digunakan sebagai data untuk sumbu `x`.

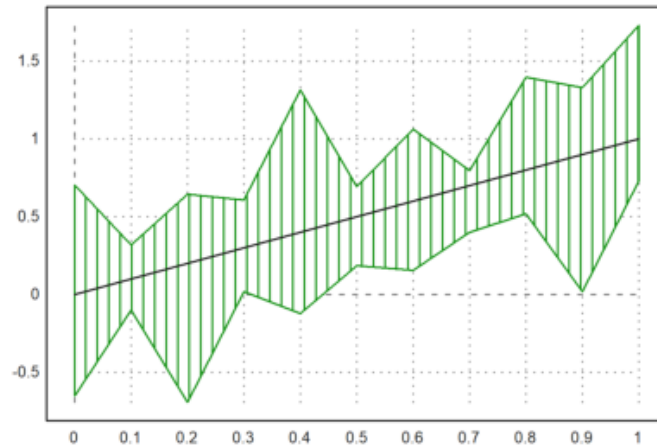
`y` adalah vektor yang digunakan sebagai data untuk sumbu `y`.

`<grid` mengaktifkan garis-garis koordinat (grid) di latar belakang plot, membantu dalam visualisasi.

`<frame` mengaktifkan bingkai (frame) di sekitar plot.

`>filled` mengisi area di bawah kurva dengan warna, membuat plot menjadi lebih berwarna

```
>t=0:0.1:1; ...
>plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|"); ...
>plot2d(t,t,add=true):
```



Penjelasan:

- $t = 0:0.1:1$;

Ini adalah perintah untuk membuat vektor t yang berisi nilai-nilai dari 0 hingga 1 dengan interval 0.1. Hasilnya adalah vektor $[0, 0.1, 0.2, 0.3, \dots, 0.9, 1]$.

- `plot2d(t, interval(t - random(size(t)), t + random(size(t))), style="|");`

Ini adalah perintah untuk membuat plot pertama. Rincian perintah ini adalah sebagai berikut:

> `interval(t - random(size(t)), t + random(size(t)))` adalah interval yang digunakan untuk menggambar "garis" pada plot. Setiap titik pada sumbu x (t) akan dihubungkan oleh dua garis vertikal yang dibuat secara acak di sekitar titik tersebut menggunakan `random(size(t))`. Hasilnya adalah plot dengan garisgaris vertikal yang mewakili interval acak di sekitar setiap titik pada sumbu x .

> `style="|"` mengatur gaya plot menjadi garis vertikal ("|").

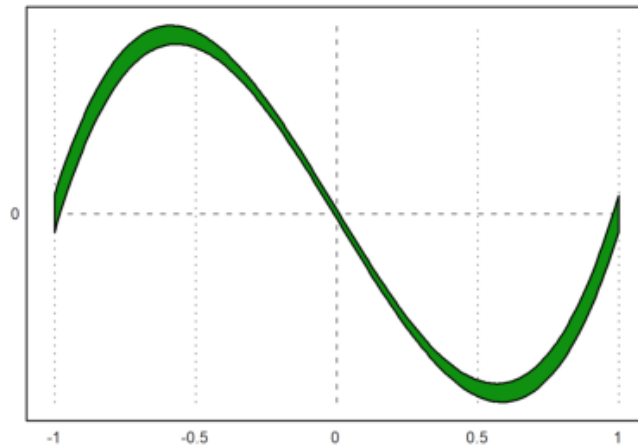
- `plot2d(t, t, add=true);`

Ini adalah perintah untuk membuat plot kedua dan menambahkannya ke dalam plot yang sudah ada dari perintah sebelumnya. Rincian perintah ini adalah sebagai berikut:

> t adalah sumbu x dan y plot ini, sehingga plot ini akan menjadi plot garis diagonal dengan kemiringan 45 derajat.

> `add=true` digunakan untuk menambahkan plot ini ke dalam plot sebelumnya, sehingga kedua plot akan ditampilkan dalam satu plot yang sama.

```
>t=-1:0.01:1; x=~t-0.01,t+0.01~; y=x^3-x;
>plot2d(t,y):
```



Penjelasan:

- $t = -1:0.01:1$;

Ini adalah perintah untuk membuat vektor t yang berisi nilai-nilai dari -1 hingga 1 dengan interval 0.01.

Hasilnya adalah vektor t yang berisi nilai-nilai seperti $[-1, -0.99, -0.98, \dots, 0.99, 1]$. Vektor t ini akan digunakan sebagai sumbu x pada plot.

- $x = t - 0.01, t + 0.01$;

Ini adalah perintah yang menghitung vektor x . Tanda digunakan di sini untuk mendefinisikan dua interval, yaitu $[t - 0.01, t + 0.01]$. Ini menghasilkan vektor x yang memiliki dua interval, satu yang kurang dari $t - 0.01$ dan satu yang lebih dari $t + 0.01$.

- $y = x^3 - x$;

Ini adalah perintah yang menghitung vektor y sebagai fungsi dari x . Fungsi ini menghitung nilai y dengan memasukkan setiap nilai x ke dalam rumus $x^3 - x$.

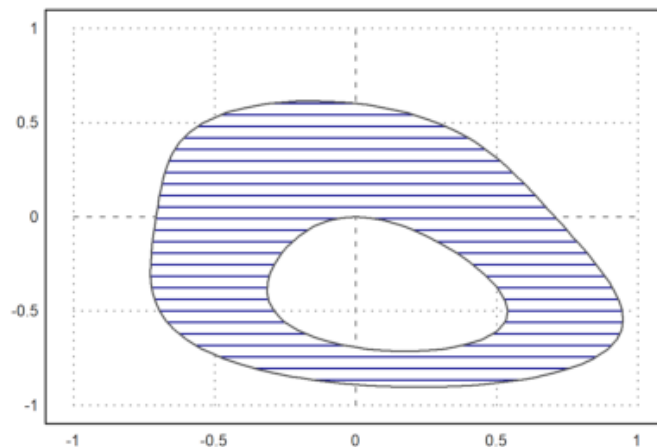
- $\text{plot2d}(t, y)$;

Ini adalah perintah untuk membuat plot dari fungsi y sebagai fungsi dari t . Rincian perintah ini adalah sebagai berikut:

> t adalah sumbu x pada plot, yang berisi vektor t yang telah didefinisikan sebelumnya.

> y adalah sumbu y pada plot, yang berisi vektor y yang dihitung dari rumus $x^3 - x$.

```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan sebuah ekspresi f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



Penjelasan:

```
- expr := "2*x^2+x*y+3*y^4+y";
```

Ini adalah perintah untuk mendefinisikan ekspresi matematika yang disimpan dalam variabel `expr`. Ekspresi ini merupakan suatu fungsi $f(x, y)$ yang tergantung pada dua variabel, yaitu x dan y . Ekspresi ini memiliki bentuk matematika yang terdiri dari berbagai suku, seperti kuadrat dari x , perkalian $x*y$, kuadrat dari y , dan lainnya.

```
- plot2d(expr, level=[0;1], style="-", color=blue);
```

Ini adalah perintah untuk membuat plot dari fungsi $f(x, y)$ yang telah didefinisikan sebelumnya. Berikut adalah rincian perintah ini:

> `expr` adalah ekspresi yang akan digunakan sebagai fungsi yang akan diplotkan. Dalam hal ini, ekspresi $2x^2 + xy + 3y^4 + y$ adalah fungsi $f(x, y)$ yang telah didefinisikan sebelumnya.

> `level=[0;1]` mengatur tingkat kontur (contour levels) yang akan digunakan dalam plot. Dalam hal ini, tingkat kontur berada pada interval 0 hingga 1, yang berarti plot akan menunjukkan wilayah di mana $f(x, y)$ memiliki nilai antara 0 hingga 1.

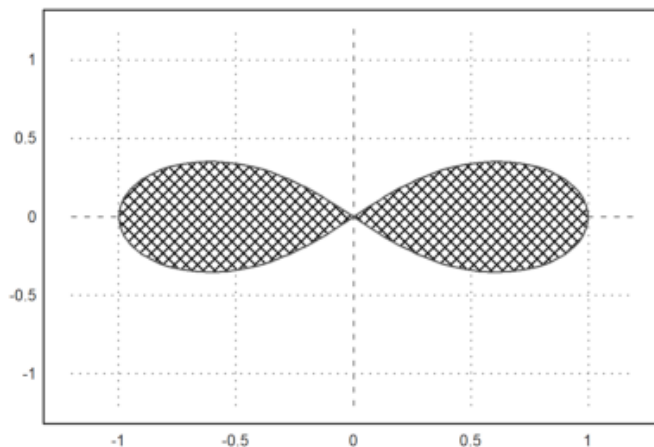
> `style="-"` mengatur gaya plot menjadi garis berjenis -, yang akan menghasilkan plot kontur.

> `color=blue` mengatur warna garis plot menjadi biru

Kita juga dapat mengisi rentang nilai seperti berikut:

$$-1 \leq (x^2 + y^2)^2 - x^2 + y^2 \leq 0$$

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/\");
```



Penjelasan:

```
- plot2d("(x^2+y^2)^2-x^2+y^2", r=1.2, level=[-1;0], style="/\");
```

Ini adalah perintah untuk membuat plot dari fungsi matematika yang didefinisikan dalam bentuk string:

$(x^2 + y^2)^2 - x^2 + y^2$. Fungsi ini tergantung pada dua variabel, yaitu x dan y .

$(x^2 + y^2)^2 - x^2 + y^2$ adalah rumus dari fungsi matematika yang akan diplotkan.

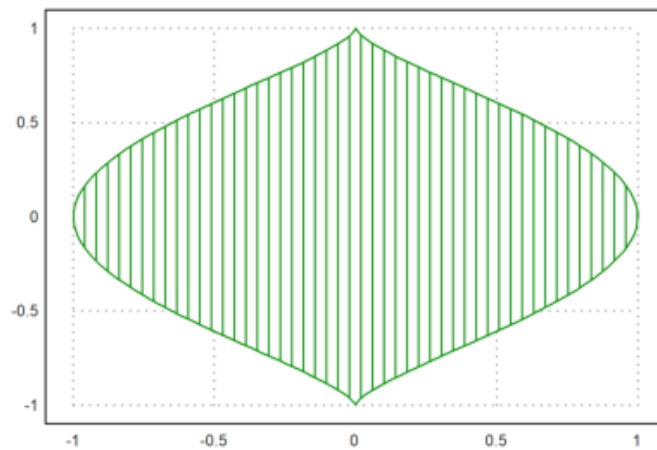
- `r=1.2` mengatur rentang (range) plot untuk kedua sumbu x dan y . Dalam hal ini, rentangnya adalah $[-1.2, 1.2]$, yang berarti plot akan berada dalam wilayah ini.

- `level=[-1;0]` mengatur tingkat kontur (contour levels) yang akan digunakan dalam plot. Dalam hal ini, ada dua tingkat kontur: -1 dan 0. Ini akan menentukan wilayah kontur dalam plot.

- `style="/\"` mengatur gaya plot menjadi garis berpotongan ("/\").

Ini akan menghasilkan plot dengan garis-garis berpotongan yang menggambarkan kontur fungsi. Di sini menggunakan kombinasi dua simbol yaitu slash dan backslash ("/\") untuk membuat garis-garis yang berpotongan.


```
>plot2d("sin(x)^3", "cos(x)", xmin=0, xmax=2*pi, >filled, style="l"):
```



Penjelasan:

```
plot2d("sin(x)^3", "cos(x)", xmin=0, xmax=2*pi, >filled, style="l");
```

Ini adalah perintah untuk membuat plot dari dua fungsi matematika, yaitu $\sin(x)^3$ dan $\cos(x)$, dalam satu plot yang sama. Berikut adalah rincian perintah ini:

- " $\sin(x)^3$ " adalah ekspresi pertama yang akan diplotkan. Ini adalah fungsi trigonometri $\sin(x)$ yang dipangkatkan tiga. Fungsi ini tergantung pada variabel x .
- " $\cos(x)$ " adalah ekspresi kedua yang akan diplotkan. Ini adalah fungsi trigonometri $\cos(x)$. Fungsi ini juga tergantung pada variabel x .
- $xmin=0$ dan $xmax=2\pi$ mengatur rentang (range) plot untuk sumbu x dari 0 hingga 2π . Ini adalah rentang yang akan ditampilkan dalam plot.
- $>filled$ mengisi area di bawah kurva fungsi dengan warna, sehingga area di bawah kurva fungsi akan diisi dengan warna.
- $style="l"$ mengatur gaya plot menjadi garis vertikal (" l "). Ini akan menghasilkan plot dengan garis-garis vertikal.

Contoh-Contoh Soal

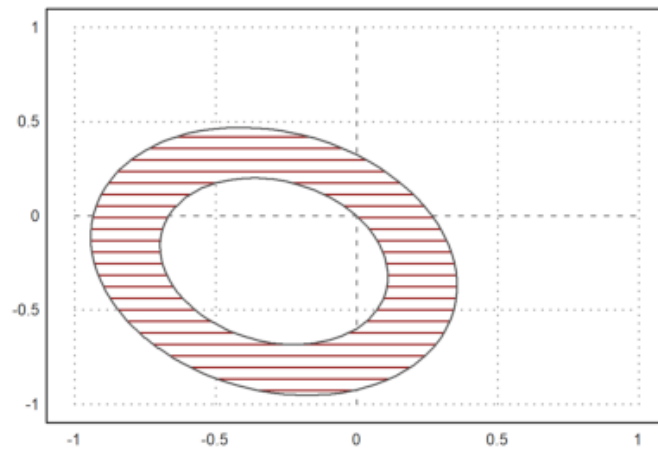
1. Gambarkan plot fungsi berikut ini

$$6x^2 + 5y^2 + 2xy + 4x + 3y$$

dengan interval

$$0 \leq f(x, y) \leq 1.5$$

```
>expr := "6*x^2+5*y^2+2*x*y+4*x+3*y"; // mendefinisikan sebuah ekspresi f(x,y)
>plot2d(expr, level=[0;1.5], style="-", color=red): // 0 <= f(x,y) <= 1.5
```



2. Gambarkan plot fungsi berikut:

$$2x^2 + 3x + 1$$

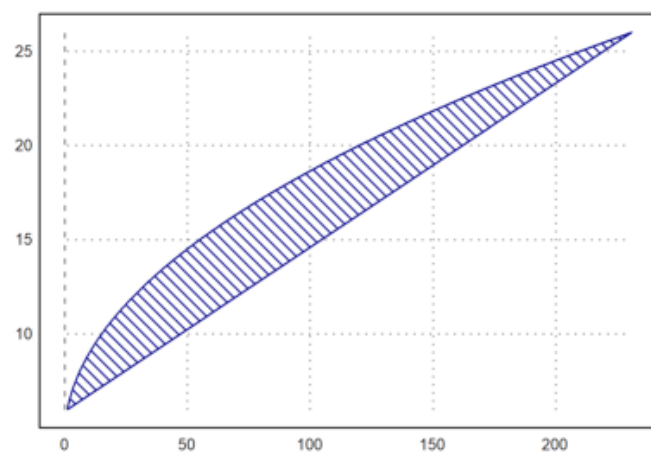
$$2x + 6$$

dengan

$$x_{min} = 0$$

$$x_{max} = 10$$

```
>plot2d("2*x^2+3*x+1","2*x+6",xmin=0,xmax=10,>filled,fillcolor=blue,style="\"):
```



3. Gambarkan plot fungsi berikut:

$$3x^2 + 3x + 1$$

$$2x + 6$$

$$-3x + 2$$

dengan

$$xmin = -10$$

$$xmax = 0$$

```
>plot2d("3*x^2+3*x+1", "2*x+6", "-3*x+2", xmin=-10, xmax=0, >filled, fillcolor=orange, style="#")
```

